

The image features two thick, black L-shaped brackets. One is positioned on the left side, with its vertical bar extending downwards and its horizontal bar extending to the right. The other is on the right side, with its vertical bar extending upwards and its horizontal bar extending to the left. These brackets frame the central text.

NODE JS EXPRESS

Node JS Express

- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- Express provides a thin layer of fundamental web application features, without obscuring Node.js features

Features of Express

- Following are some of the core features of Express framework –
 - Allows to set up middle wares to respond to HTTP Requests.
 - Defines a routing table which is used to perform different actions based on HTTP Method and URL.
 - Allows to dynamically render HTML Pages based on passing arguments to templates.

Creating an express application

- `var express=require('express');`
- `var app=express();`
- `app.get("/",function(req,res){`
- `res.send("Hello World");`
- `})`

- `var server=app.listen(8081,function(){`
- `var host=server.address().address;`
- `var port =server.address().port;`
- `console.log("app listening at "+host+" "+port);`
- `})`

Request and Response Object

- [Request Object](#) – The request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers,
- [Response Object](#) – The response object represents the HTTP response that an Express app sends when it gets an HTTP request.

Get And Post Method

- A get method can be used to access a content directly using a url.
- Post method can be used to receive data from a form

Post Example

```
■ app.post("/login",function(req,res){  
■     var user=req.body.username;  
■     var password=req.body.password;  
■     if(user=='admin' && password=='pass123')  
■     {  
■         res.redirect('/welcome.html')  
■     }  
■     else  
■     {  
■         res.redirect("/login.html");  
■     }  
■  
■ })
```

Routing in express js

- ***Routing*** refers to how an application's endpoints (URLs) respond to client requests.
- You define routing using methods of the Express app object that correspond to HTTP methods; for example, `app.get()` to handle GET requests and `app.post` to handle POST requests.

Route Parameters

- Route parameters are named URL segments that are used to capture the values specified at their position in the URL. The captured values are populated in the `req.params` object, with the name of the route parameter specified in the path as their respective keys.

Introduction ECMA Script

- ECMA stands for European Computer Manufacturer's Association
- ECMAScript is a Standard for scripting languages such as JavaScript, JScript, etc.
- It is a trademark scripting language specification
- JavaScript is a language based on ECMAScript. A standard for scripting languages like JavaScript, JScript is ECMAScript. JavaScript is considered as one of the most popular implementations of ECMAScript.

The Strict Mode in ECMA 5

- The fifth edition of the ECMAScript specification introduced the Strict Mode. The Strict Mode imposes a layer of constraint on JavaScript. It makes several changes to normal JavaScript semantics.

Var and let key word

SN	var	let
1.	The var keyword was introduced with JavaScript.	The let keyword was added in ES6 (ES 2015) version of JavaScript.
2.	It has global scope.	It is limited to block scope.
3.	It can be declared globally and can be accessed globally.	It can be declared globally but cannot be accessed globally.
4.	<p>Variable declared with var keyword can be re-declared and updated in the same scope.</p> <p>Example:</p> <pre>function varGreeter(){ var a = 10; var a = 20; //a is replaced console.log(a); } varGreeter();</pre>	<p>Variable declared with let keyword can be updated but not re-declared.</p> <p>Example:</p> <pre>function varGreeter(){ let a = 10; let a = 20; //SyntaxError: //Identifier 'a' has already been declared console.log(a); } varGreeter();</pre>
5.	<p>It is hoisted.</p> <p>Example:</p> <pre>{ console.log(c); // undefined. //Due to hoisting var c = 2; }</pre>	<p>It is not hoisted.</p> <p>Example:</p> <pre>{ console.log(b); // ReferenceError: //b is not defined let b = 3; }</pre>

Const Keyword

- The const keyword allows you to declare a constant (a JavaScript variable with a constant value).
- Constants are similar to let variables, except that the value cannot be changed.
- ```
var x = 10;
// Here x is 10
{
 const x = 2;
 // Here x is 2
}
```

# Default Parameters

- ES6 allows function parameters to have default values.
- ```
function myFunction(x, y = 10) {  
  // y is 10 if not passed or undefined  
  return x + y;  
}
```

For /of

- The JavaScript for/of statement loops through the values of an iterable objects.
- for/of lets you loop over data structures that are iterable such as Arrays, Strings

ECMA 2016 new includes

- Exponential Operator
- Array.includes